Lecture 23 Nov 16, 2010

Proof Expressions for Refinement Logic

The main pattern of the rules is "top down", 5 operators, 11 rules

	Left (Elimination)	Right (Introduction)
&	$H, A \& B, H' \vdash G$	$H \vdash A \& B$
$\vee$	$H, A \vee B, H' \vdash G$	$H \vdash A \vee B$
$\Rightarrow$	$H, A \Rightarrow B, H' \vdash G$	$H \vdash A \Rightarrow B$
$\forall$	$H, \forall x{:}A.B, H' \vdash G$	$H \vdash \forall x{:}A.B$
$\exists$	$H, \exists x{:}A.B, H' \vdash G$	$H \vdash \exists x{:}A.B$
false	H, false, $H' \vdash G$	no rule

Axiom(hyp) $H, x{:}A, H' \vdash A$
Cut
Thin
Magic

plus the Axiom, structural rules (Cut, thin), and Magic.
The standard names for the rules are opL, opR
for each of &, $\vee$, $\Rightarrow$, $\forall$, $\exists$ plus a rule for falseL.

We now introduce operators for the proof expressions.
They have this format

	Left	Right
&	spread(x; l, r. exp)	pair(a; b)
$\vee$	decide(x; l. exp_1; r. exp_2)	inl(a), inr(b)
$\Rightarrow$	apseq(x; b; y. exp)	λ(x. exp)
$\forall$	apseq(x; b; y. exp)	λ(x. exp)
$\exists$	spread(x; l, r. exp)	pair(a; b)

If we assume a single type U as the universe, then we can remove the types on the quantifiers. In λPRL we have two types, $\mathbb{Z}$ and List, so we need the typing option on the rules.

Lecture 23 continued 2

We will group the rules by the operators. First the Right side rules which introduce constructors, so called introduction rules.

Right (Constructor Introduction)

$\&$ $\quad H \vdash A \& B$ by pair(— ; —)
① $H \vdash A$ by a
② $H \vdash B$ by b

$\exists$ $\quad H \vdash \exists x{:}A.B$ by pair(— ; —)
① $H \vdash a \in A$ by a
② $H \vdash B(a)$ by $b(a)$

$\Rightarrow$ $\quad H \vdash A \Rightarrow B$ by $\lambda(x.$ — $)$ new x
$H, x{:}A \vdash B$ by $b(x)$

$\forall$ $\quad H \vdash \forall x{:}A.B$ by $\lambda(x$ — $)$ new x
$H, x{:}A \vdash B(x)$ by $b(x)$

$\vee$ $\quad H \vdash A \vee B$ by inl( )
$H \vdash A$ by a
$H \vdash A \vee B$ by inr( )
$H \vdash B$ by b

false $\quad$ no rule

Recall, $\neg A$ is $A \supset$ false in Refinement Logic.

Lecture 23 continued 3

Proof expressions continued

Left (Constructor "Elimination" or Use)

& $\quad H,\ x{:}\ A\&B,\ H' \vdash G$ by spread(x; l, r. ___)

l, r new, x given in hypotheses

$H,\ l{:}A,\ r{:}B,\ H' \vdash G$ by $g(l, r)$

(labels on hypotheses must be unique in all sequents)

x labels the hypothesis

$\exists \quad H,\ x{:}\ \exists y{:}A.\ B(y),\ H' \vdash G$ by spread(x; y, r. ___)

$H,\ y{:}\ A,\ r{:}\ B(y),\ H' \vdash G$ by $g(y, r)$

$\Rightarrow \quad H,\ x{:}\ A \Rightarrow B,\ H' \vdash G$ by apseq(x; ___ ; y. ___) y new

order not relevant, only insertion

① $H,\ x{:}A \Rightarrow B,\ H' \vdash A$ by a

② $H,\ x{:}A \Rightarrow B,\ H',\ y{:}B \vdash G$ by $g(y)$

$\forall \quad H,\ x{:}\ \forall y{:}A.B,\ H' \vdash G$ by apseq(x; a ; z. ___)

order irrelevant

① $H,\ x{:}\ \forall y{:}A.B,\ H' \vdash a \in A$

② $H,\ x{:}\ \forall y{:}A.B,\ H',\ z{:}\ B(a) \vdash G$ by $g(z)$

$\vee \quad H,\ x{:}\ A \vee B,\ H' \vdash G$ by decide(x; l. ___ , r. ___)

order irrelevant

① $H,\ l{:}A,\ H' \vdash G$ by $g_l(l)$

② $H,\ r{:}B,\ H' \vdash G$ by $g_r(r)$

false $\quad H,\ x{:}\ false,\ H' \vdash G$ by any(x)

Note in apseq(f; a; y. exp(y)) we can simplify the term to exp(ap(f;a)), since y stands for ap(f;a), the application of f to argument a.

Lecture 23 continued 4

Proof Expressions continued

Axiom (hyp) $H, x{:}A, H' \vdash A$ by hyp(x)

The hypotheses must always have unique labels.

Cut $H, H' \vdash G$ by {Cut C @ x} seq(x. ___ ; ___)

order irrelevant
① $H, x{:}C, H' \vdash G$ by $g(x)$
② $H, H' \vdash C$ by c

Thin $H, x{:}A, H' \vdash G$ by {thin} by g
$H, H' \vdash G$ by g

Declaration for T a data type $H, x{:}T, H' \vdash x \in T$ by x

Lemma Rule If there is a proof term p such that $\vdash P$ by p, then we can cut in P by the Cut rule and prove it by taking p for c in ② of the Cut rule.

We can simplify proof terms with this observation $f{:}A \Rightarrow B, x{:}A \vdash B$ by ap(f;x).

Also seq(x. exp(x) ; a) reduces to exp(a).

This can be done after the complete proof term is assembled. Thus abseq(f; a; v. exp(v)) reduces to exp(ap(f;a)).

We can use seq(c)(x. $\underline{b(x)}$; $\underline{a}$) if clearer. which reduces to $b(a)$.

Lecture 23 continued 5

Sample Proofs with Proof Expressions

1. $\vdash A \Rightarrow A$ by $\lambda(x.\_\_)$ called I

$x{:}A \vdash A$ by hyp x

Extract: $A \Rightarrow A$ by $\lambda(x.x)$

2. $\vdash A \Rightarrow (B \Rightarrow A)$ by $\lambda(x.\_\_)$ called K

$x{:}A \vdash (B \Rightarrow A)$ by $\lambda(y.\_\_)$

$x{:}A,\ y{:}B \vdash A$ by hyp x

Extract $\lambda(x.\lambda(y.x))$

3. $\vdash A \Rightarrow (B \Rightarrow C) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$ called S
by $\lambda(x.\_\_\_\_\_\_)$

$x{:}A \Rightarrow (B \Rightarrow C) \vdash (A \Rightarrow B) \Rightarrow (A \Rightarrow C)$
by $\lambda(y.\_\_\_\_\_\_)$

$x{:}A \Rightarrow (B \Rightarrow C),\ y{:}A \Rightarrow B \vdash A \Rightarrow C$
by $\lambda(z.\_\_\_\_\_\_)$

$x{:}A \Rightarrow (B \Rightarrow C),\ y{:}A \Rightarrow B,\ z{:}A \vdash C$
by $apseq(x; z; u.\_\_\_\_)$
$z{:} \vdash A$ by z
$u{:}B \Rightarrow C \vdash C$ by $apseq(y; z; w.\_\_\_\_)$
$\vdash A$ by z
$w{:}B \vdash C$ by $apseq(u; w; v.\_\_)$
$\vdash B$ by w
$v{:}C \vdash C$ by v

Extract
$\lambda(x.\lambda(y.\lambda(z.\ apseq(x; z; u.\ apseq(y; z; w.\ apseq(u; w; v.v)))))))$
This reduces to
$\lambda(x.\lambda(y.\lambda(z.\ ap(ap(x;z);\ ap(y;z)))))$
The term is called the S combinator.

Lecture 23 continued 6

Sample Proofs continued

4. $\vdash ((A \& B) \Rightarrow C) \Rightarrow (A \Rightarrow (B \Rightarrow C))$

by $\lambda(x.$ ___ $)$

$x: (A \& B) \Rightarrow C \vdash A \Rightarrow (B \Rightarrow C)$

by $\lambda(y.$ ___ $)$

$y: A \vdash (B \Rightarrow C)$

by $\lambda(z.$ ___ $)$

$z: B \vdash C$

by $apseq(x;$ ___ $; v.$ ___ $)$

$v: C \vdash C$ by v

note $\vdash A \& B$ can also go here $\vdash A \& B$ by $pair(y;z)$

Extract $\lambda(x. \lambda(y. \lambda(z. apseq(x; pair(y;z); v.v))))$

This reduces to $\lambda(x.\lambda(y.\lambda(z. ap(x; pair(y;z)))))$

where $ap(x; pair(y;z))$ applies the function x from pairs $A \times B$ to obtain a value in C.

5. $\vdash (A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \& B) \Rightarrow C)$

by $\lambda(x.$ ___ $)$

$x: A \Rightarrow (B \Rightarrow C) \vdash (A \& B) \Rightarrow C$

by $\lambda(y.$ ___ $)$

$y: A \& B \vdash C$

by $spread(y; xa, xb.$ ___ $)$

$xa: A, xb: B \vdash C$ by $apseq(x; xa; v.$ ___ $)$

$v: B \Rightarrow C \vdash C$ by $apseq(v;$ ___ $; xc.$ ___ $)$

$\vdash A$ by xa

$xc: C \vdash C$ by xc.

$\vdash B$ by xb

Extract $\lambda(x. \lambda(y. spread(y; xa, xb. apseq(x; xa; v. apseq(v; xb; xc.xc)))))$

Lecture 23 continued 7

Sample Proofs continued

6. $\vdash \exists y \forall x. R(x,y) \Rightarrow \forall x \exists y R(x,y)$
by $\lambda(h. \_\_\_)$

$h: \exists y \forall x R(x,y) \vdash \forall x \exists y R(x,y)$
by $\lambda(x. \_\_\_)$

$x:U \vdash \exists y R(x,y)$
by $spread(h; y_0, p. \_\_)$

$y_0:U,\ p: \forall x. R(x,y_0) \vdash \exists y R(x,y)$
by $apseq(p; x; v. \_\_)$

$v: R(x,y_0) \vdash \exists y. R(x,y)$
by $pair(y_0, v)$

Extract $\lambda(h. \lambda(x. spread(h; y_0, p. apseq(p; x; v. pair(y_0,v)))))$

This reduces to
$\lambda(h. \lambda(x. spread(h; y_0, p. pair(y_0, ap(p;x)))))$
since $v = ap(p;x)$.

7. $\vdash (\exists x P_x \Rightarrow C) \Rightarrow \forall x (P_x \Rightarrow C)$
by $\lambda(y. \_\_)$

$y: (\exists x P_x \Rightarrow C) \vdash \forall x (P_x \Rightarrow C)$
by $\lambda(x. \_\_)$

$x:U \vdash P_x \Rightarrow C$
by $\lambda(p. \_\_)$

$p: P_x \vdash C$ by $apseq(y; \_\_ ; xc. \_\_)$

$xc: C \vdash C$ by xc

$\vdash \exists x P_x$
by $pair(x,p)$

Extract
$\lambda(y. \lambda(x. \lambda(p. apseq(y; pair(x;p); xc. xc))))$
Reduces to $\lambda(y. \lambda(x. \lambda(p. ap(y; pair(x;p)))))$

Lecture 23 continued 8

Proof Expressions examples

8. $\vdash \neg \exists x. \neg Px \Rightarrow \forall x\, Px$

by $\lambda(h. \text{---})$

Note $\neg Q$ is $Q \Rightarrow false$ for any Q

$h: \neg \exists x. \neg Px \vdash \forall x\, Px$

by $\lambda(x. \text{---})$

$x: U \vdash Px$ by cut $Px \vee \neg Px$

by $seq(d. \text{---} ; \text{---})$

$d: Px \vee \neg Px \vdash Px$ by $decide(d; l. \text{---}, r. \text{---})$

$\vdash Px \vee \neg Px$

by $magic(Px)$

$l: Px \vdash Px$ by l

$r: \neg Px \vdash Px$

by $apseq(h; \text{---} ; f. \text{---})$

$f: false \vdash Px$ by $any(f)$

$\vdash \exists x \neg Px$

by $pair(x, r)$

~~Extract $\lambda(h. \lambda(x. seq(d. decide(d; l.l; r. apseq(h; pair(x,r); f. any(f)))))$~~

Extract $\lambda(h. \lambda(x. decide(magic(Px);$
$l.l;$
$r. apseq(h; pair(x,r), f. any(f)))))$

Where $seq(d. exp(d); magic(Px))$ reduced to $exp(magic(Px))$.

Lecture 23 continued 9

Example 9 from Smullyan p. 55

$\vdash \forall x(P_x \Rightarrow Q_x) \Rightarrow (\forall x P_x \Rightarrow \forall x Q_x)$ by $\lambda(h_1.\ \_\_\_)$

$h_1 : \forall x(P_x \Rightarrow Q_x) \ \vdash\ \forall x P_x \Rightarrow \forall x Q_x$ by $\lambda(h_2.\ \_\_\_)$

$h_2 : \forall x P_x \ \vdash\ \forall x. Q_x$ by $\lambda(x.\ \_\_\_)$

$x{:}U \vdash Q_x$ $apseq(h_1; x; v_1.\ \_\_\_)$

$x{:}U \vdash x \in U$

$v_1 : P_x \Rightarrow Q_x \vdash Q_x$ $apseq(h_2; x; v_2.\ \_\_\_)$

$v_2 : P_x \vdash Q_x$ $apseq(v_1; v_2; v_3.\ \_\_\_)$

$\vdash P_x$ by v_2

$v_3 : Q_x \vdash \cancel{P_x}\ Q_x$ by v_3

Extract $\lambda(h_1. \lambda(h_2. \lambda(x. apseq(h_1; x; v_1. apseq(h_2; x; v_2. apseq(v_1; v_2; v_3. v_3))))))$

Note $v_1 = ap(h_1; x)$

$v_2 = ap(h_2; x)$

$v_3 = ap(v_1; v_2) = ap(ap(h_1;x); ap(h_2;x)) = h_1(x)(h_2(x))$

In reduced form the extract is

$$\boxed{\lambda(h_1. \lambda(h_2. \lambda(x. h_1(x)(h_2(x)))))}$$

Compare this to the proof tree on page 55. The extract conveys the <u>computational content of the proof.</u>

Lecture 23 continued 10

Example 10 (contrapositive method)

$\vdash (\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$ $\lambda(x. \text{——})$

$x: \neg B \Rightarrow \neg A \vdash A \Rightarrow B$ $\lambda(xa. \text{——})$

$xa: A \vdash B$ by {Cut $B \vee \neg B$} seq(xd. —— ; ——)

$xd: B \vee \neg B \vdash B \vee \neg B$ by magic(B)

$\vdash B$ by decide(d; l.—, r.—)

$l: B \vdash B$ by l

$r: \neg B \vdash B$ by apseq(x; $\uparrow$; v. —)

$\vdash \neg B$ by r

$v: \neg A \vdash B$ by apseq(v; xa; f.—)

recall $\neg A$ is $A \supset$ false

$\vdash A$ by xa

f. false $\vdash B$ by any(f)

Extract

λ(x. λ(xa. seq(xd. decide(xd; l.l; r. apseq(x; r; v.
apseq(v; xa; f. any(f)))
; magic(B))))

since xd = magic(B)
v = ap(x; r)
f = ap(v; xa)

The extract then reduces to

λ(x. λ(xa. decide(magic(B); l.l; r. ~~apseq~~ ap(ap(x;r); xa)))).

Lecture 23 continued 11

Example 11 contrapositive

$\vdash (A \Rightarrow B) \Rightarrow (\neg B \Rightarrow \neg A)$ by $\lambda(h. \_\_)$

$h : A \Rightarrow B \vdash \neg B \Rightarrow \neg A$ by $\lambda(xb. \_\_)$

$h : A \Rightarrow B,\ xb : \neg B \vdash \neg A$ by $\lambda(xa. \_\_)$

$xa : A \vdash false$

by $apseq(h; \_\_; v. \_\_)$

$\vdash A$ by xa

{ recall $\neg B$ is $B \Rightarrow false$ }

$v : B \vdash false$ by $apseq(xb; \_\_; f. \_\_)$

$\vdash B$ by v

$f. false \vdash false$ by f

Extract

$\lambda(h. \lambda(xb. \lambda(xa.\ apseq(h; xa; v.\ apseq(xb; v; f.f))))))$

note ~~[illegible]~~

$v = ap(h; xa)$

$f = ap(xb; v)$

Reduced extract

$\lambda(h. \lambda(xb. \lambda(xa.\ ap(xb; ap(h; xa)))))$

Note this result gives that $\neg B \Rightarrow \neg A \Rightarrow (\neg\neg A \Rightarrow \neg\neg B)$

Show that using magic $\neg\neg A \Rightarrow A$. Without magic $A \Rightarrow \neg\neg A$.